Generalization of Data Poisoning attacks to targeted objects

Sunandini Sanyal, R. Venkatesh Babu
Department of Computational and Data Sciences
Indian Institue of Science, Bangalore, India
sunandinis@iisc.ac.in

Abstract

Security of machine learning models are a primary area of concern in today's era. Traditionally, deep learning models have been found to be susceptible to data poisoning and backdoor attacks, where attackers can modify training data to cause a particular image to be misclassified. However, previous methods of attack suffer from the shortcoming that they work only for a specific image. We propose a novel variant of the attack where an object in any pose or background can be misclassified by the model. In this work, we propose a complete optimization procedure to add small perturbations to a very small fraction of the training data to misclassify multiple images of the same object in different poses. We further show that our attack generalizes well to unseen images of the same object. We also evaluate our attacks against a state of the art defense and propose a novel optimization to make the attack more robust. We believe that our work would accelerate future research and development of novel attacks and defenses for a multi-target attack setup.

1 Introduction

In today's world, machine learning models have become popular and effective for several downstream tasks. Along with the success of machine learning, various challenges have come up regarding the security of such models. One such threat is a data poisoning attack where the attacker is able to modify the data which is used for training a machine learning model. Malicious data can get injected into victim's models via data scraping from web pages or from publicly available datasets/repositories. This causes the model to behave differently for specific images, but the overall performance of the model remains unaffected. For instance, an autonomous driving system could misinterpret a stop signal as a green signal and cause great damage, whereas it correctly identifies other traffic signs. This makes the attack very hard to detect. According to a recent Microsoft survey [10], data poisoning attacks have been identified to be a top concern in the industry. Some researchers also consider data poisoning as an important tool to understand the behaviour of machine learning models more deeply.

In this work, we focus on poisoning attacks in the transfer learning setup where a victim fine-tunes a pretrained model using malicious data. This is a more challenging setup than the from-scratch setup where the victim trains his model from scratch. The attacker adds small perturbations to a very small subset set of the training data, referred to as "poisons". When a victim unknowingly trains his model using these "poisons", the model learns to recognise specific target images as an adversarial class during inference. In our work, we focus only on "clean-label" attacks where the attacker cannot modify the ground truth labels of the data. This makes it impossible to detect poisons visually. Also, data poisoning attacks in the past literature mainly focus on maliciously misclassifying a specific target image. Such attacks are less practical when the same object appears in a different pose, background or illumination. So, we introduce a novel attack setup where multiple

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

images of the same target object can be maliciously misclassified by the model. We demonstrate that our approach can poison the model for an object in different poses with a small poison budget of 1%. Hence, our attack can generalize better in more practical scenarios. For instance, we might want to prevent a facial recognition system from identifying a particular person, even if the same individual appears in a different pose, apparel, background. The system would fail if we train it to poison only for a specific image of the person.

The contributions of this work are summarised as follows:

- We formulate the data poisoning attack problem as a multi target optimization where we want to add perturbations to a small fraction of the training data images such that the poisoned images come close to the target images in the feature space of the model. We use the Bullseye Polytope approach [1] to formulate the objective as a feature distance minimization problem.
- We propose different variants of the loss objective and show best results for two of them. On top of this, we propose a few optimizations which improves the performance of the attack over the baselines.
- We propose a novel method to construct an optimal set of poisons by selecting poisons with the minimum L2 distance from the targets in the feature space.
- We propose that LPIPS distance is more effective than L-infinity norm based distance for modelling perturbations to the poisons and show improved results for this heuristic over the baseline methods.
- We propose an "antidote optimization" to create poisons which are robust to a state of the art defense and also observe improvements in the clean validation accuracy.

2 Related Work

In the past years, different kinds of adversarial attacks and defenses have come up in the area of data poisoning. The methods can be broadly divided into two kinds of approaches : feature based and bilevel optimization. Convex Polytope [12] and Bullseye Polytope [1] are feature distance based attacks where poisons build a convex hull around the target image. The Convex Polytope uses coefficients to position the poisons around the target images, whereas in Bullseye Polytope method, we take the coefficients as equal and minimise the average feature distance of the poisons with the target image. Polytope methods are popular in the transfer learning setting, where a victim finetunes a pretrained model on a fine tuning dataset.

Other methods treat the optimization as a bilevel optimization problem where we jointly minimize the overall training loss and the cross entropy loss of the target images with respect to an adversarial class. Witches' Brew [2] solves the bilivel optimization problem by aligning the gradient updates of the model for the target images with the poison images. They demonstrate the effectiveness of the attack in a from-scratch setting where victims trains a model from scratch using the poisoned dataset.

In addition to such attacks, there are trigger based attacks [8] which insert a secret trigger in the images. This causes the model to behave maliciously for images with a trigger embedded in it. Among the data poisoning and backdoor attacks, Bullseye Polytope was found to be most in the transfer learning setting. Hence, we build our methods on top of this baseline approach.

The defenses against data poisoning attacks mainly rely on data sanitization approaches to detect and remove poisons from the dataset. One of the state of the art defense is the Deep KNN defense [7] which looks at the k nearest neighbours of every point of the dataset and removes it if it's label mismatches the mode of the labels of the k neighbours. Another defense called l2-norm Outlier Defense removes a fraction μ of points from every class c that are farthest in feature space from their centroid. In this work, we evaluate our attack against the Deep Knn defense which is better than l2-norm centroid defense.

3 Proposed loss formulation

In this section, we discuss the loss formulations that we use to incorporate multiple targets in the optimization objective. We use the insights of the Bullseye Polytope [1] attack where we want the poison samples' convex hull to surround the target image. Since there are multiple targets, we desire the poisons to get evenly distributed in the feature space, close to the targets. In the rest of the section, we would describe how to achieve this outcome.

3.1 BP-MT - Bulleseye Polytope Multi Target Loss

The Bulleye Polytope method introduced a feature based loss objective. Intuitively, the feature embeddings learnt by the model for the poison images should lie close to the targets. So we start by minimising the distance between the average feature embeddings of the poisons and the targets. $\phi^{(i)}$ denotes the feature embedding of model i. There are N target images, represented using x_t and there are k poison images denoted as x_p. x_b is the base poison image corresponding to x_p and we add perturbations to x_p to construct a poison. We sum up the feature loss over an ensemble of m models. We treat the attack as a constrained optimization problem where the perturbations added to the poison images are within an L-infinity ϵ ball. This is expressed using the L-infinity norm constraint. The loss objective is written as the following equation which serves as a very strong baseline method for our experiments.

$$min_{\{x_p^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k}\phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2},$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)})||_\infty \leq \epsilon,$$

3.2 BP-Max-Loss - Bulleseye Polytope Multi Target Max Loss

We observe that the objective function takes high values for some models where the distance in the feature space between the poisons and the targets are higher. This could also lead to a higher loss for models which have not been seen before. We want the optimizer to optimize aggressively for such models. So, we add an extra parameter which acts as a model regulariser. The coefficeint value c_i is chosen to be inversely proportional to the feature distance between poisons and targets for that particular network. This ensures that we penalize the loss more for the networks where the feature distances are higher. Hence, the loss formulation is modified to,

$$min_{\{x_p^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} c_i \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k}\phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2},$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)})||_\infty \leq \epsilon,$$

where c_i is a model regularizer.

3.3 BP-Multi-Layer-Loss - Bulleseye Polytope Multi Target Multi Layer Loss

The original formulation of the Bullseye Polytope attack did not take into consideration different layers of the CNN model. From past literature on CNN visualizations [11], we know that different layers in a CNN model are capable of capturing features from different parts of an image. The lower layers focus on more local features, whereas the higher layers capture features for the entire image. So including lower layers helps in the optimization further and we modify the loss objective as follows,

$$min_{\{x_p^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \sum_{l=1}^{L} \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i,l)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k}\phi^{(i,l)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2},$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)})||_\infty \leq \epsilon,$$

where L is the number of layers of the ith CNN model.

3.4 BP-Feature-Cluster-Loss - Bulleseye Polytope Multi Target Feature Cluster Loss

The presence of multiple target objects makes the attack difficult since we want to place poisons close to each of the targets in the feature space. The target images might themselves be far away from each other in the feature space, especially if we choose two different models of a car to poison for. Hence, we divide the target feature space in k number of cluster, where k is the number of poisons. Every target image gets assigned to its closest cluster head and distance of the poisons with the cluster head is minimised. This helps the optimization to place each poison close to a group of target images. Hence, the BP-MT objective is optimised as follows for the clusters.

$$min_{\{x_p^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \frac{\frac{1}{k}\sum_{j=1}^{k} ||\phi^{(i)}(x^{(j)}_{t-cluster-head}) - \phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N} \phi^{(i)}(x_t^{(j)})||^2},$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)})||_\infty \leq \epsilon,$$

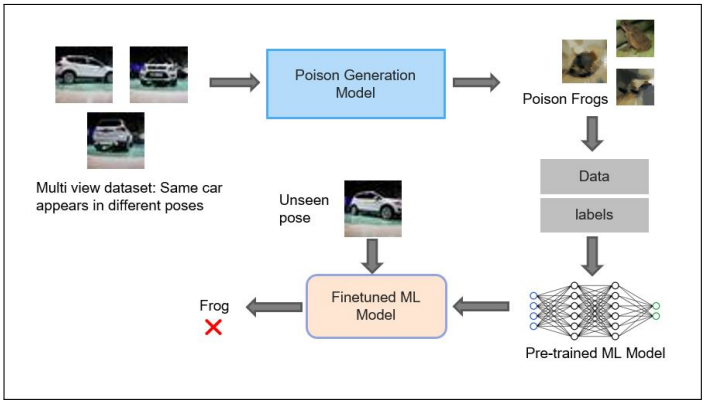

Figure 1: Poison Generation Setup: Images of the same car in different poses are used to generate poisons using feature distance optimization. The optimization produces clean frog images from CIFAR-10 as poisons which are injected into the fine tuning dataset. The attack is succesful when during inference the model is able to recognise the car's image as a frog. The accuracy is measured against both seen and unseen poses of the car.

4 Poison Generation Setup

The poisoning attack accuracy is measured in a transfer learning setup where a victim fine tunes a pretrained model on a poisoned dataset. For our experiments, we consider models pretrained on CIFAR-10 dataset. The finetuning set consists of 500 images, which are drawn from CIFAR-10. The poison budget is only 1% of the training data, which is 5 in this case. For our experiments, the poisons are created using an ensemble of three models - ResNet50, DPN92, and SENet18. An ensemble of models leads to a better feature representation and more attack efficiency.

For getting target images of the same object in a different pose, we use the Multi-view Cars dataset [6] from EPFL lab where images of 20 cars are taken by rotating the car after every 3-4 degrees. These images are converted to CIFAR-10 resolution and are used to train the poison generation model as shown in the figure 3 The Poison generation module is trained by the attacker using 20% of the images collected for a specific car. The rest 80% of the car images are treated as unseen images and we measure the strength of the attack in terms of both the seen and unseen target images. The Poison Generation module uses an optimization procedure to generate poisons which are injected maliciously into the victim's training data. After the victim finetunes the pretrained model with the poisoned dataset, we measure the attack accuracy over 20 trials, one for each car and aggregate the attack accuracy over these trials. The details of the results are described in the next section.

Table 1: White-Box results, averaged for RestNet50 and DPN92.

Approach	Poison Acc.(%)	Unseen Poison Acc.(%)	PreTest Acc.(%)	PostTest Acc.(%)
BP-MT	30.348	32.152	92.87	91.82
BP-Max-Loss	30.649	31.394	92.87	91.96
BP-Multi-Layer-Loss	41.270	42.167	92.87	91.75
BP-Feat-Cluster-Loss	26.330	27.532	92.87	91.81

Table 2: Black-Box Results, averaged for GoogLeNet and MobileNetV2.

Approach	Poison Acc.(%)	Unseen Poison Acc.(%)	PreTest Acc.(%)	PostTest Acc.(%)
BP-MT	21.798	19.492	90.92	90.99
BP-Max-Loss	24.304	24.019	90.92	91.02
BP-Multi-Layer-Loss	31.190	31.143	90.92	90.96
BP-Feat-Cluster-Loss	20.045	18.162	90.92	91.02

5 Analysis of loss variants

We conduct experiments to examine which of the loss functions have a meaningful impact for the given problem setup. The poisons are created using an ensemble of three models, ResNet50, DPN92 and SENet18. The poison budget used is of 1%. Perturbations are added so that each pixel in the poison image is not altered by more than 0.1 epsilon value. We conduct white-box experiments to measure the model's attack effectiveness in scenarios when the same model is used for making the poisons and training on the poisoned data. The results show that BP-MT model is able to attack 30.348% of the seen target images were used for training. The model is also able to identify unseen poses of the same object which is indicated by an unseen poison accuracy of 32.152% in Table 1. The clean validation accuracy of the model is not affected much by the poisoning attack. From our black box experiments, we see that the attack is effective in a transfer setting where a different model is used for making the poisons and testing the attack. The poisons trained using an ensemble of ResNet50, DPN92 and SENet18 models are able to transfer well to other models such as GoogLeNet and MobileNetV2 as indicated in Table 2. However, the attack accuracy is lower than the white box models.

The BP-MT attack serves as a strong baseline for poisoning. The feature-cluster based objective is not effective since it is a loss formulated based on a presumption that the target images are distributed far apart in the feature space. This hypothesis may not be true for images of the same car in different poses. However, this kind of attack has scope for further research where we can try to poison for images from different classes and might turn out to be very effective. The BP-Max-Loss based objective performs well for black box attacks shown in Table 2 when the optimization tries to penalise more for models whose loss is higher. The BP-Multi-Layer loss objective performs the best among all the others since it incorporates features from all the layers of a CNN model and is more accurate in terms of capturing features from the targets and poison images. The Multi Layer loss formulation results in a high poison accuracy of 41.270% on white-box attacks and a poison accuracy of 31.190% for black-box attacks. In the following sections, we use the BP-MT and BP-Multi-Layer Loss objectives which perform the best so far to optimize the formulation further.

6 LPIPS optimization

In the BP-MT optimization, we use L-infinity norm constraints to perturb poison images within an epsilon boundary. Hence, each pixel cannot be altered by more than an ϵ limit. The perceptual similarity between poison images with respect to the original images can also be measured using other distance metrics such as SSIM, PSNR, etc. Ideally, we want the poison images to look the same as before. From the past work on adversarial attacks by Feizi [5], we infer that LPIS is a good distance metric to calibrate perceptual similarity. LPIPS calculates the L2 norm of the distance between two images in the feature space of pretrained networks such as ResNet, AlexNet and SqueezeNet. We use AlexNet for our analysis since the distance calculated using Alexnet closely resembles the way humans perceive images. It is also observed as shown in Figure 2 that after perturbing images using

LPIPS, we get poisons that visually look unchanged. Hence, we optimise the BP-MT loss objective to include the lpips distance constraint as follows,

$$min_{\{x_p^{(j)}, x_{t'}^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \frac{||\frac{1}{N}\sum_{j=1}^{N} \phi^{(i)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k} \phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N} \phi^{(i)}(x_t^{(j)})||^2},$$
$$subject\ to\ lpips - dist(x_p^{(j)}, x_b^{(j)}) \leq \epsilon,$$

In order to compute the projection of the modified poisons onto the lpips ball constraint space, we use the bisection algorithm from [5]. For every iteration of adding perturbation to poisons, the bisection method is executed to get the projection as shown in Algorithm 1. After using LPIPS, we show in the results that it improves over the original BP-MT attack.

Figure 2: Comparison of poisons created by the BP-MT attack vs BP-MT + LPIPS attack. The poisoned frog images look visually same and this confirms that LPIPS is a good perceptual metric.

Algorithm 1 Perceptual Projection (Bisection Method)

procedure PROJECT(lpips threshold ϵ, adversarial poison x_p, poison base input x_b)
  $\alpha_{min}, \alpha_{max} \leftarrow 0, 1$
  $\delta \leftarrow x_p - x_b$
  while $i\ in\ 1, 2, ...n$ **do**
    $\alpha \leftarrow (\alpha_{min} + \alpha_{max})/2$
    $x^{'} \leftarrow x_b + \alpha\delta$
    if $d(x_b, x^{'}) > \epsilon$ **then**
      $\alpha_{max} \leftarrow \alpha$
    else
      $\alpha_{min} \leftarrow \alpha$
  return $x^{'}$

7 Optimal set of poisons

The BP-MT optimization is also dependent on the quality of poisons that we choose for the optimization procedure. Hence, we make an attempt to choose the set of poison images optimally. We want to choose poisons that are close to the boundary of the poison class so that the objective can be optimized faster with a fewer iterations. Hence, the poisons should be close to the target images in the feature space. The feature space is computed from the penultimate layer of the models and L2 norm distance is used as a metric to measure the distance in the feature space. This heuristic shows promising results which have been described in the results section.

8 Results of optimizations on BP-MT

From our previous analysis, we use the BP-MT and BP-MT Multi layer loss formulations which are found to be most effective and apply the lpips distance approach to them. LPIPS optimization gives a significant improvement over BP-MT, especially in ResNet50. The optimal set of poisons are selected from the nearest ones to the target images and this method is referred to as "nearest" in the

results table. This approach results in improvements in all the white box and black box models. The clean validation accuracy remains unaffected by the attack in all the experiments.

Table 3: White-Box results - DPN92

Approach	Attack Accuracy	Unseen Attack Acc	Pre-poison test acc	Post-poison test acc
BP-MT	24.096	23.019	92.920	92.684
BP-MT+LPIPS	28.718	29.416	92.920	92.944
BP-MT + Nearest	47.641	44.484	92.920	92.919
BP-MT + Nearest + LPIPS	58.800	57.590	92.920	92.918
BP-MT-Multi	42.767	42.083	92.920	92.996
BP-MT-Multi+LPIPS	54.354	52.538	92.920	93.059
BP-MT-Multi + Nearest	57.232	53.091	92.920	93.022
BP-MT-Multi + Nearest + LPIPS	72.532	69.427	92.920	92.995

Table 4: White-Box results - ResNet50

Approach	Attack Accuracy	Unseen Attack Acc	Pre-poison test acc	Post-poison test acc
BP-MT	27.795	28.478	92.820	91.755
BP-MT+LPIPS	50.911	51.195	92.820	90.577
BP-MT + Nearest	55.021	52.704	92.820	91.554
BP-MT + Nearest + LPIPS	63.168	61.843	92.820	90.870
BP-MT-Multi	68.437	63.296	92.820	90.458
BP-MT-Multi+LPIPS	63.860	60.772	92.820	90.479
BP-MT-Multi + Nearest	72.368	67.774	92.820	90.764
BP-MT-Multi + Nearest + LPIPS	74.403	74.026	92.820	90.424

Table 5: Black-Box results - GoogLeNet

Approach	Attack Accuracy	Unseen Attack Acc	Pre-poison test acc	Post-poison test acc
BP-MT	19.223	19.480	92.110	92.328
BP-MT+LPIPS	43.474	40.722	92.110	92.720
BP-MT + Nearest	48.778	47.535	92.110	92.429
BP-MT + Nearest + LPIPS	53.236	48.527	92.110	92.188
BP-MT-Multi	31.811	32.264	92.110	92.752
BP-MT-Multi+LPIPS	52.706	53.247	92.110	92.358
BP-MT-Multi + Nearest	59.688	58.999	92.110	92.665
BP-MT-Multi + Nearest + LPIPS	71.440	70.262	92.110	92.645

9 Optimization on Defenses

In order to increase the overall robustness of the model and prevent the clean accuracy from dropping further, we introduce an optimization strategy that can make the model resilient to the existing defenses. Most of the existing defenses against data poisoning attacks make an attempt to remove the poisons from the training dataset by using certain heuristics. For example, the Deep-KNN defense makes an attempt to remove poisons from the data by examining the labels of the closest neighbours. For every point in the training data, k nearest neighbours are picked up and, if the label of the current data point matches the mode of the labels of the k neighbours, then the data point is benign, else it gets filtered as a poison. Such defense strategies are effective, especially when the data points get accurately modeled in the feature space.

We propose an optimization which makes the attack more robust to such defenses. From our previous discussions, we have seen that the BP-MT attack adds perturbations to the poison images such that it comes close to the target images in the feature space. Also, in the feature space, there could be many other images from the target class which are close to the target images. For example, if the targets are from "car" class, there could be many other images in the "car" class. So we pick up images from the target class that are close to the target images using euclidean distance in the feature space and refer to them as "antidotes". These antidote images are potential images which cause the defense algorithm to detect the poisons. So we would want to push away the antidote images from the target images, so that after the BP-MT attack, when the poison images from the poison class come close to the target images in the feature space, the antidote images would not be

Table 6: Black-Box results - MobileNetV2

Approach	Attack Accuracy	Unseen Attack Acc	Pre-poison test acc	Post-poison test acc
BP-MT	15.784	15.179	89.730	89.522
BP-MT+LPIPS	15.929	16.008	89.730	89.557
BP-MT + Nearest	30.417	30.073	89.730	89.371
BP-MT + Nearest + LPIPS	35.452	33.221	89.730	89.359
BP-MT-Multi	18.663	19.616	89.730	89.515
BP-MT-Multi+LPIPS	27.350	25.734	89.730	89.491
BP-MT-Multi + Nearest	39.189	39.040	89.730	89.361
BP-MT-Multi + Nearest + LPIPS	51.025	48.585	89.730	89.354

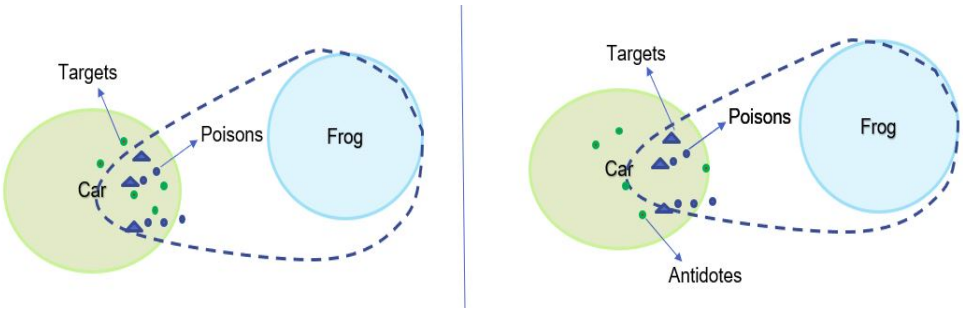

Figure 3: Comparison of BP-MT attack vs BP-MT along with antidotes. The antidote images are perturbed so that they move away from the target images in the feature space and the poisons come close to the target images.

detected as k nearest neighbours for the poisons. This also helps to increase the clean validation accuracy by 0.39% since we are making the boundary between the poison and the target class even more distinct. This behaviour is depicted in Figure 3

The following loss formulation is used to perturb the poison images as well as the antidote images. The antidote images are perturbed such that the feature distance between antidotes and targets are maximised. So a regularizer is added to the BP-MT objective to enforce the constraint as follows,

$$min_{\{x_p^{(j)}, x_{t'}^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k}\phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2} - \lambda \frac{1}{2m} \sum_{i=1}^{m} \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)}) - \frac{1}{k'}\sum_{j=1}^{k'}\phi^{(i)}(x_{t'}^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2}$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)}||_\infty \leq \epsilon,\ ||x_t^{(j)} - x_{t'}^{(j)}||_\infty \leq \epsilon$$

where t' denotes the index of antidote images from the target class and the λ is the regularization constant.

Instead of using the feature based loss, we can perturb the antidote images such that the cross entropy loss between the antidote images and the target class label is minimised. So the following loss formulation is also used as means to achieve the same goal.

$$min_{\{x_p^{(j)}, x_{t'}^{(j)}\}} \frac{1}{2m} \sum_{i=1}^{m} \frac{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)}) - \frac{1}{k}\sum_{j=1}^{k}\phi^{(i)}(x_p^{(j)})||^2}{||\frac{1}{N}\sum_{j=1}^{N}\phi^{(i)}(x_t^{(j)})||^2} - \lambda \frac{1}{2m} \sum_{i=1}^{m} \frac{1}{k'} \sum_{j=1}^{k'} Cross{-}Entropy(x_{t'}^{(j)}, y_t)$$
$$subject\ to\ ||x_p^{(j)} - x_b^{(j)}||_\infty \leq \epsilon,\ ||x_t^{(j)} - x_{t'}^{(j)}||_\infty \leq \epsilon$$

The results for the antidote optimization are shown in table where we observe that the antidote optimization results in lesser reduction in the poison accuracy over seen and unseen target images for the DPN92 network for the cross-entropy loss objective. The original BP-MT objective without any antidotes perform better for MobileNetV2 and Resnet50. The feature based loss is seen to be performing better for the both DPN92 and GoogleNet, since it leads to a lesser reduction in

the poison accuracy. In general, the antidote approach leads to a reduced accuracy of poisoning, making the attack less effective, but it increases robustness for DPN92 and GoogLeNet. In Table 7, RedSeenPoisonAcc column is reduction in the seen poison accuracy after defense algorithm is applied and Pre-DefenseSeenPoisonAcc is the poison accuracy before applying defense. Also, we know that the approach leads to better boundary seperation between the two classes. Hence, the validation accuracy overall improves by 0.39%.

Table 7: Deep-KNN Defense on BP-MT attack

Model	Pre-DefenseSeenPoisonAcc.(%)	RedSeenPoisonAcc.(%)	RedUnSeenPoisonAcc.(%)
DPN92	46.770	61.665	53.331
GoogLeNet	44.786	57.456	52.294
MobileNetV2	30.161	43.650	37.182
ResNet50	62.515	63.196	60.845

Table 8: Deep-KNN Defense on BP-MT with antidote attack with Cross entropy Loss

Model	Pre-DefenseSeenPoisonAcc.(%)	RedSeenPoisonAcc.(%)	RedUnSeenPoisonAcc.(%)
DPN92	39.141	49.790	52.086
GoogLeNet	40.695	59.355	53.842
MobileNetV2	27.931	60.929	56.332
ResNet50	58.754	70.906	70.920

Table 9: Deep-KNN Defense on BP-MT with antidote attack with Feature Distance Loss

Model	Pre-DefenseSeenPoisonAcc.(%)	RedSeenPoisonAcc.(%)	RedUnSeenPoisonAcc.(%)
DPN92	45.805	59.757	57.0577
GoogLeNet	40.962	51.581	50.702
MobileNetV2	28.691	53.845	58.004
ResNet50	60.246	71.072	73.886

10 Conclusion and Future Scope

In this work, we evaluate the effectiveness of data poisoning attacks in a practical scenario where we have multiple target images which we want to poison for. We propose a novel set up for data poisoning and a suitable baseline approach of using Bullseye Polytope attack. We also propose new loss formulations on top of the baseline approach and demonstrate the effectiveness of the attack in a transfer setting for a variety of deep learning models. We further propose two optimizations, LPIPS and selecting an optimal set of poisons, which make the attack stronger. In order to guarantee robustness of the model against defenses, we propose an optimization based on antidotes and observe a reasonable robustness. This helps us to build a robust and efficient poisoning attack.

Our work is one of the first attempts to make poisoning attacks generalizable so that a specific object in any background, pose, appearance can be identified by the model. There is further need of research in this direction where we can explore more on the challenges of making these attacks generalize to any background or situation. For instance, we could employ a multi-class multi-target approach where multiple images from different classes can be poisoned together in a single trial. This would also stimulate research to build defenses against such attacks.

References

[1] H. Aghakhani, D. Meng, Y.-X. Wang, C. Kruegel, and G. Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. *arXiv preprint arXiv:2005.00191*, 2020.

[2] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

[3] J. Geiping, L. Fowl, G. Somepalli, M. Goldblum, M. Moeller, and T. Goldstein. What doesn't kill you makes you robust (er): Adversarial training against poisons and backdoors. *arXiv preprint arXiv:2102.13624*, 2021.

[4] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein. Data security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

[5] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

[6] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 778–785. IEEE, 2009.

[7] N. Peri, N. Gupta, W. R. Huang, L. Fowl, C. Zhu, S. Feizi, T. Goldstein, and J. P. Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In *European Conference on Computer Vision*, pages 55–70. Springer, 2020.

[8] A. Saha, A. Subramanya, and H. Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965, 2020.

[9] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

[10] R. S. Siva Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75, 2020.

[11] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[12] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623. PMLR, 2019.

11 Ablations

In this section, we analyse the impacts of the various parameters on our experiments. The choice of parameters plays a crucial role in determining the balance between the effectiveness and practicality of the attack. For instance, the victim may not scrape all images from the web when they are using malicious data to train their models. Hence, we need to ensure that attacks are effective with a smaller set of poisons. Also, selection of parameters such as epsilon, poison budget play a huge role in the way the attack behaves. Some of these findings are illustrated in the following subsections. We use car model 17 from the Multi-view cars dataset for all our ablations with 5 poisons on a finetuning dataset of 500 images and use a pretrained ResNet50 model to measure to attack accuracy.

11.1 Effect of poison budget

The poison budget is the amount of training data that can be altered by an attacker. We usually prefer to choose a very small budget of around 1 %. It can be seen from Table 10 that as we increase the poison budget, the attack accuracy increases. However, we show that our attack is robust enough as it performs well with a small poison budget.

11.2 Effect of Epsilon

The epsilon value is the limit by which we can alter each pixel of the poison images, We observe from Table 11 that as we increase the epsilon, attack accuracy increases. But if we make the epsilon too high, victims might be able to visually inspect images and find the poisons as they would look much different with respect to their original image, So it is important to trade-off accuracy with perceptual similarity.

Table 10: Impact of poison budget on BP-MT attack performance on ResNet50 with car model 17 and fixed epsilon=0.1

Poison Budget	Attack Acc.%	Unseen Attack Acc.%
1%	32.631	36.818
2%	56.315	42.727
4%	64.211	56.363

Table 11: Impact of epsilon on BP-MT attack performance on ResNet50 with car model 17 and fixed budget=1%

Poison Budget	Attack Acc.%	Unseen Attack Acc.%
0.03	6.265	6.815
0.06	8.965	7.570
1.00	32.631	36.818

11.3 Effect of increasing seen target images

As we increase the number of target images that we want to poison for, the efficiency of the attack seems to be reducing. The attack accuracy on unseen target images seem to reduce as well. This means that along with memorising the seen target images, the model is able to generalize well to unseen target images. Hence, the model surprisingly shows better generalization for lower seen percentage of target images as shown in Table 12.

Table 12: Impact of increasing targets on BP-MT attack performance on ResNet50 with car model 17 and fixed budget=1%

Seen Targets %	Attack Acc.%	Unseen Attack Acc.%
5%	83.333	95.121
10%	83.333	73.504
25%	65.388	69.902
50%	55.813	61.627

11.4 Comparison with gradient alignment

The Witches' Brew paper introduced a novel approach to align gradient updates of the poisons with the targets. We compare this method with the BP-MT objective for car model 17 and average over two trials. We use the L2 distance between the gradients instead of the cosine distance. As described in the paper [3], we observe similar results that this attack is weaker than the Bullseye Polytope attack in the transfer learning setting. In Table 13 we see that the BP-MT attack gives a better attack accuracy.

11.5 Case Study : Demonstration of the attack on CIFAR-100 for a person's image

We take the Bullseye Polytope attack and examine the effective of the attack in a practical scenario where we train the attack to poison for different images of a famous person "Obama" standing in different poses, background and apparel. We use a ResNet50 model, pretrained on CIFAR 100 dataset and poison 5 images from class "Bee". During inference, we see that 20% of the images get misclassified into class "Bee". This indicates that our model is partly effective in practical scenarios and there is need for further research in this direction.

Table 13: Comparison of Witches Brew with BP-MT attack performance on ResNet50 with car model 17 and fixed budget=1%, epslion=0.1

Approach	Attack Acc.%
BP-MT	73.33
Grad-Alignment	70.00